Information Visualization and Interaction

Debajyoti Majumdar
M.S in Data Science

ABSTRACT

Information is facts provided or learned about something or someone, or the knowledge obtained from investigation, study, or instruction [1]. A StackOverFlow Developer Survey data-set has been explored to extract information, gain insights by representing it using various visualization techniques. Multiple interaction techniques have been explored to support data analysis. The report gives an insight to various data analysis that has been done and the results that have been found.

Index Terms: Information Visualization—StackOverFlow Survey data-set—Visualization techniques—Dashboard;

1 INTRODUCTION

Information Visualization has been used for extensive research for many years to get sensible interpretation of the available raw data as information expands faster than our ability to comprehend it. Visualization makes data easier to understand through direct sensory experience (usually visual), as opposed to more linguistic/logical reasoning [2]. Here, we have designed an interactive dashboard for showcasing the visualizing techniques for analyzing the StackOverFlow Survey data-set. For creating the dashboard we have used the following technologies:

- Python - for data pre-processing and creating the dashboard. Libraries for supporting charts and interaction:
  - plotly
  - pymysql
  - dash
  - pandas
- MySQL Workbench - for supporting the back-end

The remainder of the report is organized as follows:
Section 2 contains the problem description, the list of tasks and questions from an analyst's point of view and an overview on how we go about the analysis of the data. Section 3 discusses about the various visualization techniques and the motivation of our choice over a set of them. In Section 4, we explain and summarize our findings during data-set exploration and we have captured the same in a screen cast. Section 5 covers the evaluation of our approach and we summarize our contributions and findings in Section 6.

2 PROBLEM DESCRIPTION AND TASK ANALYSIS

2.1 Problem Details

In this report, we had to select a data-set of our own choice and analyze it by framing questions from a data analyst's point of view. Next, to answer these questions we had to transform the data into a suitable format and use various visualization techniques and present it on an interactive dashboard to provide a clear picture to the end user about the data-set.

2.2 Tasks

Brief overview of the tasks that were undertaken are given below:

(a) To *find an information data-set* which was sufficiently large and complex.

After exploring numerous data-sets we came across the **StackOverFlow Developer Survey for 2018** [3]. This was sufficiently complex with more than 98K participants (rows) and 129 attributes (columns).

(b) In order to *work with the data-set*, we could take one of the two methods:

- Process the data at file level
- Load it into a database

We decided to go ahead with the second option, as it would be more convenient to play around with the data and also help us choose the right set of questions that we wanted to answer. From an analysts perspective, we wanted the data in a clean format, which would be much easier to achieve when dealing with databases. We have used MySQL database for storing our data-set.

(c) Next step is to perform Data Pre-processing. This is an essential step from an analysts perspective, as any type of noise can affect the insights. The following tasks were performed to clean the data:

- Removing of extra white-spaces
- Converting the text data to similar case
- Removing special characters

(d) Now that we have the clean data-set, we explore the data-set thoroughly. During this phase we made a list of questions from an analysts perspective. Refer Section 2.3 for the same.

(d) Since the scope of this data-set was huge we prioritized the questions that an analyst would want an answer to, and shortlisted the questions. The shortlisted questions have been mentioned in the section below.

(e) Based on each question, we make a probable list of visualization techniques that would best suit in representing the data and later finalize on the best one based on user experience. Also, the type of interaction was decided based on how the analyst would prefer to interact with the application and how the interaction would support in exploring the data-set. This has been explained in detail in 3.

(f) A *interactive dashboard* application is *developed* which contains all the different visualization technique integrated together to give an overview of the data-set.

We made a screencast which covers the interaction and integration of visualization techniques.

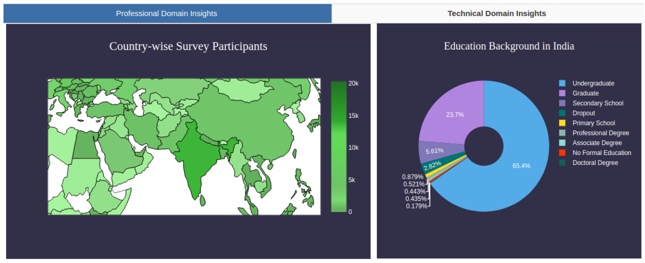


Figure 1: Screenshot of the Dashboard

(g) We use this application (dashboard) to answer the questions and *discuss our observations* and *evaluate our approach*.

Refer to Section 4 for the observations and Section 5 for the evaluation.

(h) Finally, we *summarize our contributions and findings*.

Refer to Section 6 for our contributions and findings.

2.3 Questions Framed

Below are the list of questions that we have decided to answer: -

1. How many participants have filled up the survey? Which countries have been **most/least involved** in this survey?

2. What is the educational background of the participants from **a specific country**?

3. How many of the participants are satisfied with their current job and how many are open to new opportunities? Do we see any relation between the these two attributes?

4. In a specific country, what are the number of hours that a developer spends on computer and how many hours does he/she spend outside? Do we see any work-life balance here?

5. From the entire population of participants of a specific country, what are the different types of role types that have been worked in?

6. For a specific developer in a country, what are the 10 most/least frequently used programming languages for the Year 2018? What are the most/least frequent programming languages that these developers desire to work in 2019?

7. For a specific developer in a country, what are the 10 most/least frequently used databases for the Year 2018? What are the most/least frequent databases that these developers desire to work in 2019?

8. For a specific developer in a country, what are the 10 most/least frequently used platforms for the Year 2018? What are the most/least frequent platforms that these developers desire to work in 2019?

2.4 Attributes Used

Out of the 129 attributes present in the data-set, we selected the following:

- Country
- Formal Education
- DevType
- JobSatisfaction
- JobSearchStatus
- LanguageWorkedWith
- LanguageDesireNextYear
- DatabaseWorkedWith
- DatabaseDesireNextYear
- PlatformWorkedWith
- PlatformDesireNextYear
- HoursComputer
- HoursOutside

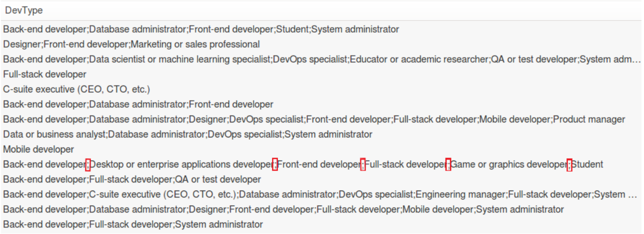
2.5 Data Analysis

On framing the questions we needed to develop solutions for answering these questions. We faced challenges while answering each question. Simple database operations (selection, projection and aggregation) were not enough to achieve the results. Below we have mentioned the steps that were undertaken, along with the operations performed in order to answer some of these questions.

(a) For Question 3 in (Refer Section 2.3), we have tried to think outside the box and find analyze if there is any relation between the current job satisfaction vs job searching status. We put on an analysts hat and asked ourselves this question.

(b) For Question 4, we wanted to see the average time that the developers are spending on computer vs the time spent outside. On comparing the visualization of Question 3 and 4, do we see any type of relation? Does it vary from highly developed countries to under-developed countries? These type of questions go way beyond simple aggregation of data.

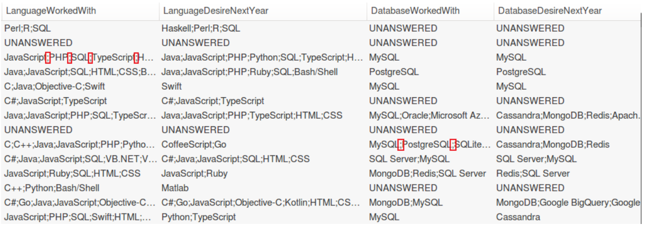
(c) For Question 5, we needed to use the *DevType* attribute which was one of the multi-valued columns. From figure 2, we can see that the developer has been a (mention the roles here). Therefore while we aggregate the data at a developer role level, we need to consider them in the bucket for all the above mentioned roles. To achieve this, we split the attribute value based on the separator (;). Next step was to include this participant in each of this role.



DevType
Back-end developer;Database administrator;Front-end developer;Student;System administrator
Designer;Front-end developer;Marketing or sales professional
Back-end developer;Data scientist or machine learning specialist;DevOps specialist;Educator or academic researcher;QA or test developer;System adm…
Full-stack developer
C-suite executive (CEO, CTO, etc.)
Back-end developer;Database administrator;Front-end developer
Back-end developer;Database administrator;Designer;DevOps specialist;Front-end developer;Full-stack developer;Mobile developer;Product manager
Data or business analyst;Database administrator;DevOps specialist;System administrator
Mobile developer
Back-end developer;Desktop or enterprise applications developer;Front-end developer;Full-stack developer;Game or graphics developer;Student
Back-end developer;Full-stack developer;QA or test developer
Back-end developer;C-suite executive (CEO, CTO, etc.);Database administrator;DevOps specialist;Engineering manager;Full-stack developer;System …
Back-end developer;Database administrator;Designer;Front-end developer;Full-stack developer;Mobile developer;System administrator
Back-end developer;Full-stack developer;System administrator

Figure 2: DevType Attribute from the StackOverFlow Survey dataset

(d) For question 6, 7 and 8 we used *LanguageWorkedWith* & *LanguageDesireNextYear*, *DatabaseWorkedWith* & *DatabaseDesireNextYear* and *PlatformWorkedWith* & *PlatformDesireNextYear* and applied a similar operation as Question 5. However, since there are over a 100 technologies that a developer has worked in, we decided to display the top 10 and bottom for each category. There is no attribute in the dataset that can help us derive this insight directly. Hence this operation goes beyond the possibilities of the provided data.



LanguageWorkedWith	LanguageDesireNextYear	DatabaseWorkedWith	DatabaseDesireNextYear
Perl;R;SQL	Haskell;Perl;R;SQL	UNANSWERED	UNANSWERED
UNANSWERED	UNANSWERED	UNANSWERED	UNANSWERED
JavaScript;PHP;SQL;TypeScript;H…	Java;JavaScript;PHP;Python;SQL;TypeScript;H…	MySQL	MySQL
Java;JavaScript;SQL;HTML;CSS;B…	Java;JavaScript;PHP;Ruby;SQL;Bash/Shell	PostgreSQL	PostgreSQL
C;Java;Objective-C;Swift	Swift	MySQL	MySQL
C#;JavaScript;TypeScript	C#;JavaScript;TypeScript	UNANSWERED	UNANSWERED
Java;JavaScript;PHP;SQL;TypeScr…	Java;JavaScript;PHP;TypeScript;HTML;CSS	MySQL;Oracle;Microsoft Az…	Cassandra;MongoDB;Redis;Apach…
UNANSWERED	UNANSWERED	UNANSWERED	UNANSWERED
C;C++;Java;JavaScript;PHP;Pytho…	CoffeeScript;Go	MySQL;PostgreSQL;SQLite…	Cassandra;MongoDB;Redis
C#;Java;JavaScript;SQL;VB.NET;V…	C#;Java;JavaScript;SQL;HTML;CSS	SQL Server;MySQL	SQL Server;MySQL
JavaScript;Ruby;SQL;HTML;CSS	JavaScript;Ruby	MongoDB;Redis;SQL Server	Redis;SQL Server
C++;Python;Bash/Shell	Matlab	UNANSWERED	UNANSWERED
C#;Go;Java;JavaScript;Objective-C…	C#;Go;JavaScript;Objective-C;Kotlin;HTML;CS…	MongoDB;MySQL	MongoDB;Google BigQuery;Google
JavaScript;PHP;SQL;Swift;HTML;…	Python;TypeScript	MySQL	Cassandra

Figure 3: LanguageWorkedWith and DatabaseWorkedWith Attributes from the StackOverFlow Survey dataset

3 Visualization Design

3.1 Choice of techniques

(a) For question 1: The following visualization techniques were considered:

- Pie chart
- Bar chart
- Tree map
- Population Density Map

The problem with the first two options was that over 150 countries were involved in the survey, so the number of elements would be too much for the chart to handle and it would look too clumsy. A tree map would have been a viable option, although the size of the boxes in the tree map are proportional to the quantity of measurement. Hence, countries where the number of participants were extremely small would have been almost be invisible, thereby making the user feel that such countries had not participated in the survey at all. However, we wanted to project all the countries that had been involved in the survey. Hence the last option (***Population Density Map***) fits our needs perfectly.

(b) For question 2: The following visualization techniques were considered:

- Pie chart
- Bar graph
- Variable width column chart

In this question we wanted to show the total share of the formal education that the participants of the selected country had received. In this scenario we have few values of the variable of interest. Hence, all the above options were ideally perfect for our interest. On experimentation, the pie chart was most visually appealing. Hence, we considered a ***pie chart*** for our dashboard.

(c) For question 3: The following visual visualization techniques were considered:

- Heat map
- Scatter plot
- Scatter plot bubble size

Through this question we wanted to analyze whether the developers are satisfied with their jobs or not. And also, if their current job satisfaction affects their choice of looking for another job. Since here we had more than two variables involved, a simple bar chart or pie chart would not suffice. A scatter plot or a bubble plot would give us a better understanding on the relationship between current job satisfaction. However, heat maps has the power of making the end user focus on the important issue just by a glance. The color schemes used in the map play an important role. In this scenario we wanted to make use of the colors for making the user focus on the correlation between the current job satisfaction vs job search status. Hence, using a ***heat map*** is thereby justified.

(d) For question 4: The following visualization techniques were considered:

- Line Chart
- Bar Chart Horizontal

Here we wanted to compare how many hours a developer spends on a computer vs how many hours the developer spends outside. The comparison of these two attributes gives a clear indication on the work-life balance that a developer has in his daily life. Since we have different time buckets involved in our data, a ***line chart*** would be the best visual representation to track the changes over time.

(e) For question 5: The following visualization techniques were considered:

- Pie Chart
- Tree Map
- Scatter bubble plot with different bubble size

For this question, we wanted to showcase the variety of developers that have filled out the survey along with their density distribution. Pie chart would again not be the best choice here as we have more than 20 different developer types in our data set. The Tree map suits our needs completely as it can show the breakdown of the entire population into these categories. The second tab of our dashboard is majorly focusing on the trending technologies that are used by these developers. Hence, we have added a click interaction with the ***tree map***.

For this question, we have also incorporated a ”**click**” interaction in the treemap. When we click on one of the developer specializations, the visualization of the remaining charts in Tab 2 of the dashboard alter accordingly to give the Top 10 and Bottom 10 technologies for that particular specialization.

(f) For question 6: The following visualization techniques were considered:

- Group Bar Chart
- Stacked Bar Chart
- Pyramid Bar Chart

For this question, we wanted to show the comparison of the most frequently used technologies by the developers of a specific domain and how these technologies perform in the year 2019 among them. A bar chart is ideal for this type of scenario. The real challenge was which type of bar chart to choose. On experimentation with the above mentioned options, we realized that the ***group bar chart*** made perfect sense and was easy to compare the performance of these technologies among various domains.

(g) For question 7 and 8, we used the same reasoning as above and decided to settle with group bar charts, as we were ideally answering similar type of questions.

3.2 Modifications made to existing code

(a) *Population Density (or Choropleth) Map -*

- colorscale - here we have defined our own customized color scale. By using the default conditions, we were not able to differentiate between the participants population among countries which were at a similar level. By defining our own color scale, we could detect minute changes in the participants population, especially in the African continent. [4]

(b) Pie/Donut chart -

- colorscale - here we have defined our own colorscale for each section of the donut chart. This has been further discussed in the later sections. [5]

(c) Heatmap and Treemap -

- colorscale - we have defined our own color scale here. The darker colors correspond to the higher frequencies and the lighter ones corresponds to lower frequencies. This color shading has been kept consistent throughout the dashboard, to improve user experience. [6] [7]

In this project, we created a dashboard which is interactive in nature. In this dashboard, we created two tabs with each tab having its own combination of filters for better visualization of the data. Also, we have inter-linked the visualizations in a particular tab with each other.

In Tab 1, we visualized the questions 1 through 4. Here, with the help of the choropleth visualization technique, we can hover over the each country and through which, a filter (here ”country”) gets activated and the visualization of the remaining three change with respect to the country.

In Tab 2, we have visualized the questions 5 through 8. In this tab, we have a drop down menu to select a country for which we would like to see the specialization of developers and ’top-10’ or ’bottom-10’ technologies which the developers are currently working with and want to work with in the future.

4 Use Case

4.1 Interaction

We have designed two tabs for our dashboard and interaction is present in both of them.

In the Tab 1, we have multiple charts, all of which do not fit together in the view window, hence we represented the 4 visualizations in a 2x2 format. Since we have to scroll down to look at the remaining two visualizations, we appended the ”country name” to the title of each chart in order to enhance the user experience. The country name is obtained from the hover interaction in the choropleth map which is clearly depicted in the screencast. In this tab, there is only one filter, which is obtained by the interaction. By default, India is selected.

As there are 4 charts in the second tab as well, we use a similar approach as in tab 1 and we appended the ”developer specialization” to the title of each chart to increase user’s readability which is obtained by the click interaction in the treemap. This interaction is clearly shown in the screencast. In this tab, there are two filters involved, one for the country via a drop down which is applicable to all the charts and and another for the developer specialization via a click interaction in the treemap which is applicable too all the bar charts. Also, there is an additional functionality of radio button option through which, we can view either ”Top10” or ”Bottom10” of all the technologies.

4.2 Observations and Insights

4.2.1 *”Professional Domain Insights”* Tab

- It can be observed that the aggregate of graduates and undergraduates in developed and developing nations are significantly more in comparison to the same aggregate in under-developed nations. From this observation it can be inferred that there might a possibility that the number of opportunities available for developers in developed and developing nations are more in comparison to the opportunities available in under-developed nation. Refer Figure 4.

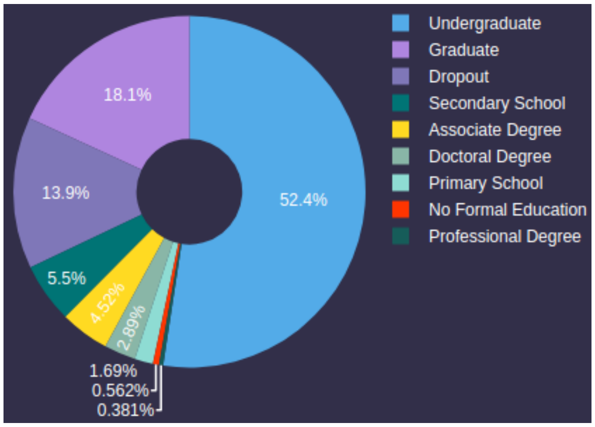


Figure 4: High Percentage of Grads and Undergrads in the US

- The percentage of dropouts in African countries are significantly higher when compared to the percentage of dropouts from a developed or a developing nation. From this we can infer that the opportunities available in these countries might be less due to them being under-developed. Refer Figure 5.

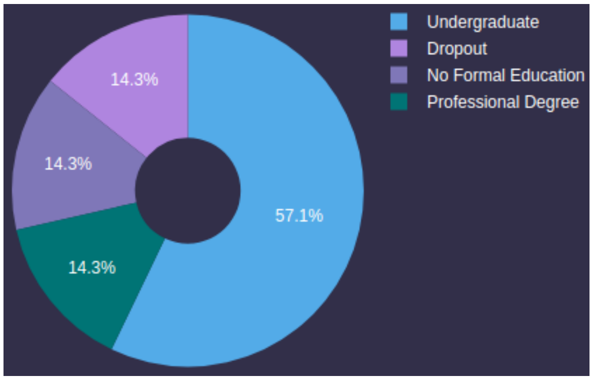


Figure 5: High Dropout Percentage in Congo

- For the North American continent, we observed that there are significant number of developers who despite being moderately or extremely satisfied with their current job, are still open to opportunities. We might say that the developers from this region are more open to new challenges. Refer Figure 6.

- In Asian countries, the observation is that the majority of developers are open to opportunities irrespective of whether

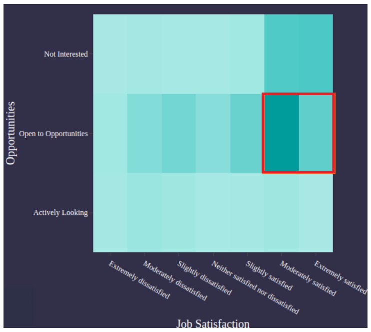


Figure 6: Developers open to Job Opportunities in Canada

they are satisfied or dissatisfied with their current job. It can be inferred that the developers from this region might be more inclined towards learning or might be looking for a change in skill. Refer Figure 7.

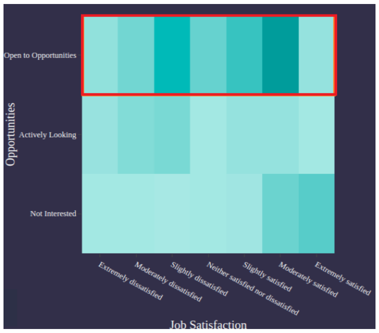


Figure 7: Developers open to Job Opportunities in Japan

- In Asian countries, it can be observed that there are very few developers who have a good work-life balance. The work-life balance gets affected as the time spent on Computer increases while the time spent outside decreases. It can be inferred that the youth population (graduates and undergraduates) which is very high in these countries might tend to work more in order to acquire more knowledge in comparison to other developers of other age groups. Refer Figure 8.

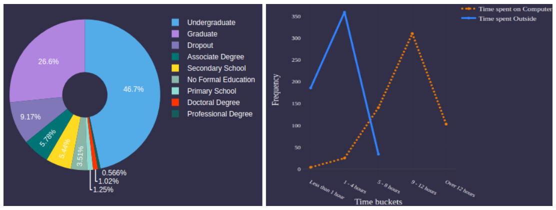


Figure 8: Work-Life Balance of Youth population in Iran

- We can observe that there are very few developers who spend 1 - 4 hours in front of the computer. We might say that they are seasoned or experienced developers who might be running a company or might be supervisors of other developers who are doing the actual development. Refer Figure 9.

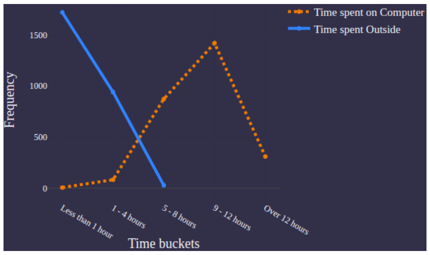


Figure 9: Work-Life Balance in Canada

- For South America and Australia, the observation is that there are small portion of developers who are extremely dissatisfied with their current jobs but still are not interested to search for a new job or look for new opportunities. The inference is that there might be other factors influencing these developers to stay at the same job. These reasons may be personal or professional or both. Refer Figure 10.

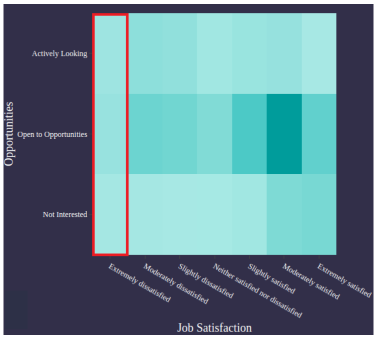


Figure 10: Job Search Status of Extremely dissatisfied developers in Brazil

- In European Union, the number of dropouts are significantly high even though almost all countries in Europe are either developed or developing one. We can infer that the developers may be the presented with a lot of opportunities to showcase their skill and take it to a next level without having to pursue a mainstream education.Refer Figure 11.

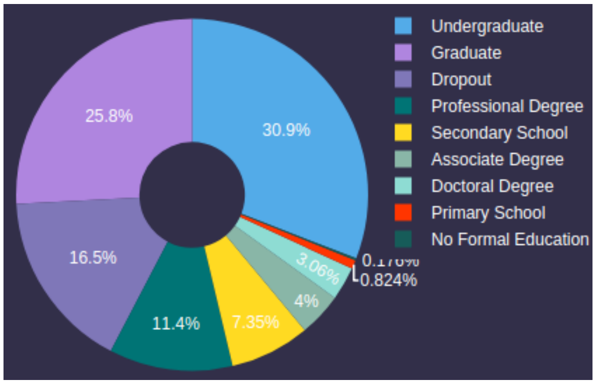


Figure 11: Significant Dropout rate of 16.5% in Spain

- The percentage of dropouts in India is very less in comparison with the rest of the world. The inference is that there might be parental or societal pressure on the youth to get a degree and earn a living. Refer Figure 12.

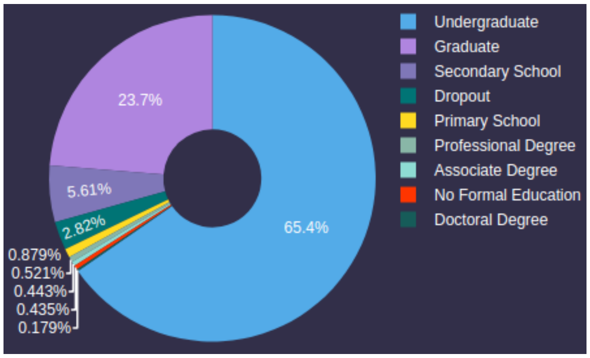


Figure 12: Lowest Dropout rate of 2.82% in India

4.2.2 *"Technical Domain Insights"* Tab

- We see that in almost all countries, there are a major portion of Back-End, Front-End and Full-Stack developers which leads us to believe in the possibility that most of the developers in all countries would have started their careers with these specializations and later branched out to expand their horizons. Refer Figure 13.

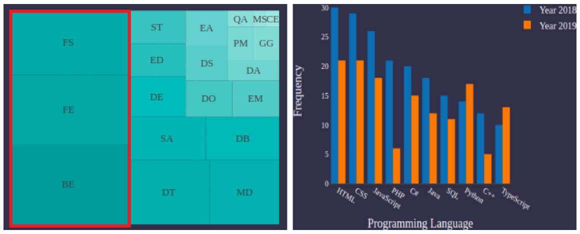


Figure 13: Most common developers and decrease in trend of top 10 languages used by them in Cuba

- It can be observed that the programming languages like Javascript, HTML and CSS appear in the Top 10 languages that the developers have worked with in most of the countries. It can be inferred that since most of the developers would have started their careers as Back-End, Front-End or Full-Stack developers, there might be a possibility that they would have worked with these languages during that phase. Refer Figure 13.

- We observe that there are developers in many countries who desire to work with Kotlin and Rust programming languages in 2019. This is evident by the increase in trend observed in 2019 when compared to 2018. We infer that the developers might be interested to learn or implement these languages due to the fact they are simpler or less prone to errors in comparison to C, C++ or Java. Refer Figure 14.

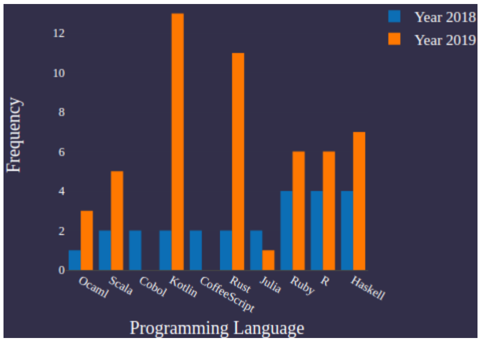


Figure 14: Increase in trend for Kotlin and Rust in Italy

- It can be seen that many developers from many countries want to work with Cloud based DBs like Google Cloud Storage, Cassandra, etc, in 2019 in comparison to physical storage DBs like MySQL, SQL Server, etc. From this, it can inferred that this might be due to the fact that the Cloud based DBs are more robust, do not run out of space and support Big Data technologies & its analysis. Refer Figure 15.

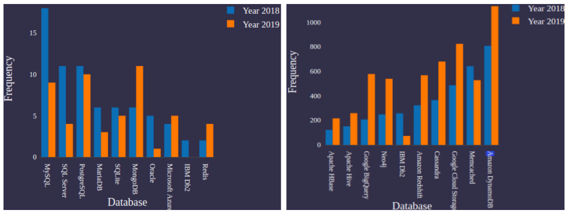


Figure 15: Decrease in trend for MySQL, SQLServer, SQLite DBs in Chile and increase in trend for Cloud Based DBs in the US

- In few of the South American countries, MongoDB and Microsoft Azure are trending among Data or business analysts. We can say that it might be based on the fact that these servers are cloud based and help the developers by performing faster analysis. Refer Figure 16.

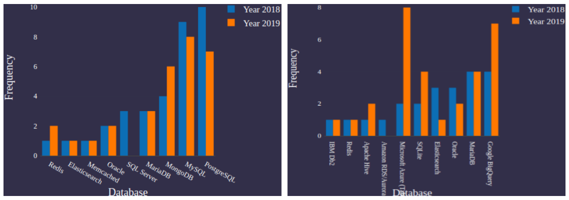


Figure 16: Increase in trend for MongoDB in Venezuela and Microsoft Azure in Colombia

- For the year 2019, the Game or Graphics Developers in few of the Asian countries are interested in working with the programming language GO. This leads us to infer that the trend shift towards GO might be because it has better garbage collection when compared to C. Refer Figure 17.

- In few of the European countries, there is a decrease in trend for platforms like Windows Desktop, iOS, Linux and Android for most specializations of developers. We can infer that the developers might be shifting towards home assistant based

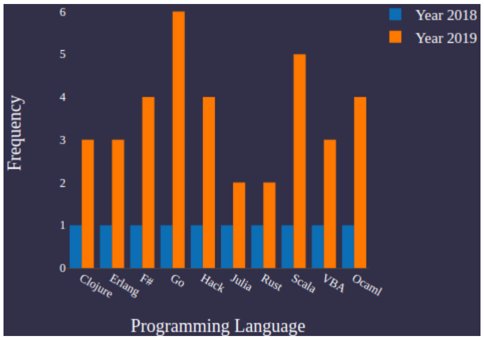


Figure 17: Increase in trend for GO among Game Developers in Bangladesh

platforms like Google Home, Amazon Echo, etc as they are currently trending in the market. Refer Figure 18.

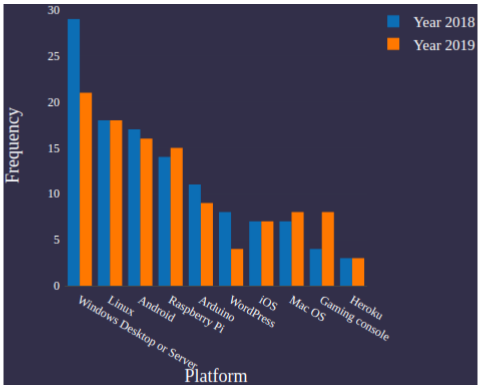


Figure 18: Decrease in trend for platforms among Game Developers in Austria

- We know that a developer has worked on multiple roles over their professional career, hence we have ended up with observations for certain developers which did not make sense entirely. For example: we observed that for Database Administrator, there is an increase in trend for platforms like Google Home, Amazon Echo, Apple Watch, etc. We inferred that even though they are completely related to each other, this might have occurred because of the fact that these developers might have worked on different platforms earlier in their career.Refer Figure 19.

5 Evaluation

While designing the dashboard we have made numerous iterations in our approach and made multiple changes in our design which are discussed in this section.

(a) At first the population density map had a flat projection (Refer Figure 20) in which the relative area of each country is approximated. In order to have a better representation, we used a ***mollweide*** projection (Refer Figure 21).

The first color-scale was Viridius which gradients between yellow and blue. But in our data-set, the number of participants vary from 1 to over 20k which makes it hard to distinguish between countries where the participation is less than 1k as the color code for these countries was the same. Hence, we changed the color-scale to a single color-gradient of Persian-green to better accommodate our data-set. We used a shade of green as it is a universal color and is visually appealing to the human eye. [8]

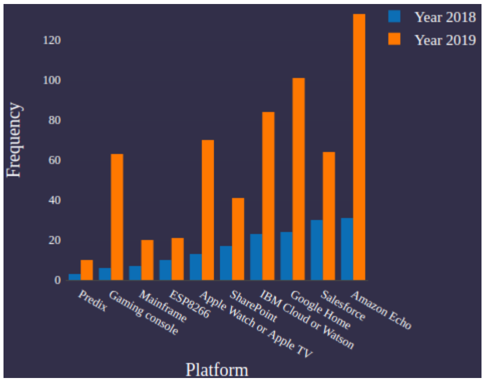


Figure 19: Platforms decreasing in trend for DB Admin in India

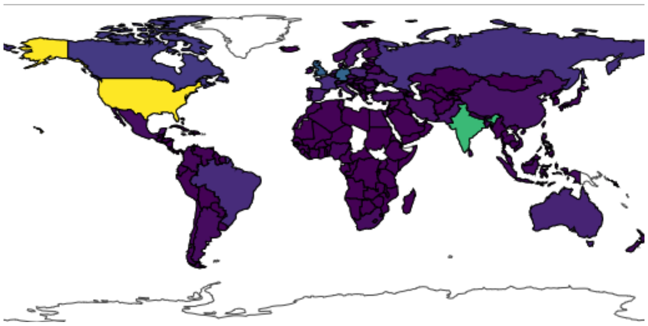


Figure 20: Flat Projection

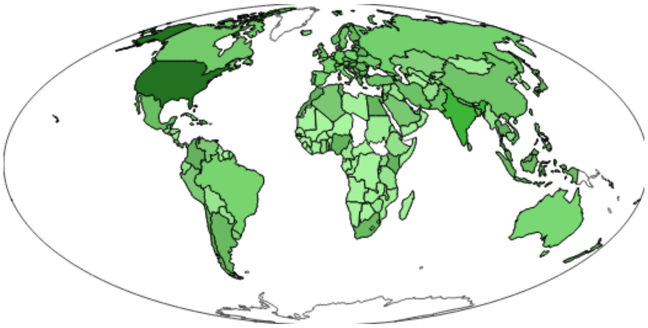


Figure 21: Mollweide Projection

(b) The default color-scale for the pie chart was very luminous. In order to make it better visually, we used our own color-scale which contained very mellow colors (in comparison) so that each portion of the pie chart would stand out clearly over the dark background.

(c) The default color-scale of the heatmap was Viridius and due to this, we were not able to draw any proper inference from it.

So, we changed the color-scale having gradient from a darker shade of Persian-green to a lighter one. Once this change was implemented, we were clearly able to tell the difference between various cells due to the intensity variation. We chose this shade of green as this was aesthetically pleasing. [9] [10]

When the first version of the treemap was implemented using different colors and it was not very appealing and led to a confusion while deciding which specialization had the lowest frequency just by looking at it. When we implemented the same color-schema as the heatmap, we were able to easily identify which specialization had the least frequency.

(d) For the line chart, the default colors were red and blue on a light grey background. The red color was too bold and was drawing attention away from the blue line. Hence, we decided to change the color-scale to blue and orange. Upon implementing this on a dark background, the line chart became more visually appealing to the eye and we were able to see the trends very clearly.

(e) During our initial design, the background for each visualization was light grey. As the screen was already white, the visualizations did not look very expressive or good. So, we shifted to a darker blue background and all our visualizations became more appealing and looked very expressive.

(f) To maintain consistency across the entire dashboard, the font size of the chart titles were kept bigger than the font size axis titles and the font size of the axis titles bigger were kept bigger than font size of the axis labels. This increased the readability of the dashboard.

The font color was white for all chart titles, axis titles, axis labels and legends. The white color on the dark background gave the dashboard visualizations a professional appearance.

(g) In our initial application design, we were displaying only top 10 languages, databases and platforms for the selected developer. On playing around with the application, we realised that displaying the top 10 would not necessarily capture the upcoming technologies. Maybe there are situations where the technologies are not so trending in 2018, but the developers want to work with them next year. To capture this perspective, we decided to add the least 10 technologies and see how they perform next year. After adding this functionality we realised that our hypothesis was right. Adding this functionality gave a new depth to our analysis.

6 Conclusion, Contribution and Future Work

At the end of our analysis, we can conclude the following from our dataset-

- There are multiple factors influencing the developers to look for new job opportunities. Job Satisfaction is not the only reason.
- Majority population of the developers are working more than the prescribed limit (8 hours/day) which is not good in general as it affects their work-life balance as well as their mental health.
- There are significant number of developers who spend less time outside which is not good either.
- Using the "Bottom 10" option in the Technical Domain Insights tab we noticed that there is an increase in trend across all technologies for 2019 leading us conclude that there are a lot of developers who are not working on that technology currently want to shift towards these upcoming ones in 2019.

For future work, if this dashboard is given to a data analyst, they can use it to ask themselves the following questions -

- The reasons why the upcoming technologies are trending in 2019. Is this the same trend across the world or is it in a specific geographic location?
- The other major reasons influencing the developers to look for new job opportunities. If not job satisfaction then what else?
- The main reasons for significant dropout rates in many developed countries.
- Main reasons why the top 10 trending technologies in 2018 are decreasing in demand in 2019.

References

[1] https://www.merriam-webster.com/dictionary/information.

[2] https://www.semanticscholar.org/paper/Introduction-to-information-visualisation-Dix/8f5aa50e4cb12e294af534096d08bf6e6ab57188.

[3] https://www.kaggle.com/stackoverflow/stack-overflow-2018-developer-survey.

[4] https://plot.ly/python/choropleth-maps/.

[5] https://plot.ly/python/pie-charts/.

[6] https://plot.ly/python/heatmaps/.

[7] https://plot.ly/python/treemaps/.

[8] http://www.b-eye-network.com/newsletters/ben/2235.

[9] https://coolors.co/.

[10] T.-M. Rhyne. Applying color theory to digital media visualization. 2017.